PRU

# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/842,387 | 04/25/2001 | Daniel D. Grove | SIA-P047 | 6789 |

7590 06/16/2003

FERNANDEZ & ASSOCIATES, LLP
PATENT ATTORNEYS
PO BOX D
MENLO PARK, CA 94026-6204

| EXAMINER | |
|---|---|
| TRUONG, CAM Y T | |
| ART UNIT | PAPER NUMBER |
| 2172 | 2 |

DATE MAILED: 06/16/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

PRG

| | | | |
|---|---|---|---|
| **Office Action Summary** | **Applicati n N .**<br>09/842,387 | **Applicant(s)**<br>GROVE ET AL. | |
| | **Examin r**<br>Cam-Y T Truong | **Art Unit**<br>2172 | |

***-- The MAILING DATE of this communication appears on the c ver sheet with th correspondence address --***

**P riod for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)□ Responsive to communication(s) filed on _____ .

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) 1-48 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-48 is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)□ The proposed drawing correction filed on _____ is: a)□ approved b)□ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)□ The oath or declaration is objected to by the Examiner.

**Priority under 35 U.S.C. §§ 119 and 120**

13)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____ .

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)□ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) □ The translation of the foreign language provisional application has been received.

15)□ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) □ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) □ Interview Summary (PTO-413) Paper No(s). _____ .
5) □ Notice of Informal Patent Application (PTO-152)
6) □ Other: _____ .

## DETAILED ACTION

1. Claims 1-48 are pending in this Office Action.

### *Claim Rejections - 35 USC § 103*

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-45 and 47-48 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mark ALLen Weiss, "Algorithms, Data Structures, and Problem Solving With C++", (hereinafter "Weiss").

Claims 1, Weiss teaches the claimed limitations:

"a head representing a first pointer to a first leaf node; a tail representing a second pointer to a second leaf node" as a left binary tree in fig. 17.10, the rood node has two pointers, the first pointer points to left leaf node. The second pointer points to right leaf node. The first pointer of root node is presented as a head. The second pointer of root node is presented as a tail (page 516, lines 1-22). Weiss fails to teach the claimed limitation "a body, logically adjacent to the head and to the tail, having a set of pointers pointing to contiguous empty nodes". However, Weiss teaches as shown in fig. 17.10, a left binary tree has seven nodes, each node has two pointers to point to their leaf nodes. Several efficient implementations of priority queues use trees, BinaryNode (): left (NULL), right (NULL). The above information shows that each leaf nodes can be empty node, which stores no data (page 516, lines 1-22; page 517, lines 11-17).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Weiss's teaching of a left binary tree has seven nodes, each node has two pointers to point to their leaf nodes. Several efficient implementations of priority queues use trees, BinaryNode (): left (NULL), right (NULL) in order to insert new data in tree or sort data tree following user's desire.

As to claim 2, Weiss teaches the claimed limitation "data of the same type" as (page 517, lines 23-60).

As to claim 3, Weiss teaches the claimed limitation "wherein the nodes form a sorted tree structure" as (page 642, fig. 20.1).

As to claim 4, Weiss teaches the claimed limitation " wherein the nodes are indexed" as (page 642, fig. 20.1).

As to claim 5, Weiss teaches the claimed limitation " wherein each leaf node comprises a number of data segments" as (page 517, lines 23-60).

As to claim 6, Weiss teaches the claimed limitations:

"a sorted tree structure" as shown in fig. 20.3 on, the left tree, which is a heap, is sorted by following the heap order property. This property indicates parent node <= child (page 642);

"an inserting of a data segment into the tree structure" as (page 647). Weiss fails teach the claimed limitation "a redistributing of empty tree nodes by employing a data structure, which enables a more rapid insertion of the data segments". However, Weiss teaches that the each node of the binary tree corresponds to an element of the array that stores the value in the node. Fig. 20-12 shows the routines that add items into the heap. The Toss routine is short; it just adds the new element x in next available location. Insert implements the percolate up using a very tight loop. The for loop that begins at 31 is (x <array [hold/2]; Hole /=2) increments the current size and sets the hole, which is represented as empty node, to the newly added node. The system iterates as long as the item in the parent node is larger than x. Line 32 moves the item in the parent down into the hole, and then the third expression in the for loop moves the hole up to the parent. When the loop terminates, line 33 places x into the hole. The above information shows that the system generates an array for each binary tree to redistribute any node in the binary tree for inserting new item in correct position. Whenever the system wants to insert many items, the system adds many holes as empty nodes into tree by using an array as a data structure to redistribute nodes . An array requires that some operation use linear time. Thus, the array can allow a more rapid insertion of items in a binary tree (page 641; pages 647-648; page 639, lines 19-22).

It would have been obvious to a person of an ordinary skill in the art at the the time the invention was made to apply Weiss's teaching of each node of the binary tree corresponds to an element of the array that stores the value in the node. Fig. 20-12 shows the routines that add items into the heap. The Toss routine is short; it just adds the new element x in next available location. Insert implements the percolate up using a very tight loop. The for loop that begins at 31 is (x <array [hold/2]; Hole /=2) increments the current size and sets the hole, which is represented as empty node, to the newly added node. The system iterates as long as the item in the parent node is larger than x. Line 32 moves the item in the parent down into the hole, and then the third expression in the for loop moves the hole up to the parent. When the loop terminates, line 33 places x into the hole for saving time during inserting data in a tree.

As to claim 7, Weiss teaches the claimed limitation "wherein the data segments may be inserted in any order" as (page 551, lines 8-21).

As to claim 8, Weiss teaches the claimed limitation "wherein the tree structure comprises non-leaf and leaf nodes" as (page 551, lines 8-21).

As to claim 9, Weiss teaches the claimed limitation "wherein the tree nodes are indexed" as (page 641, fig. 20.1).

As to claim 10, Weiss teaches the claimed limitation "wherein each leaf node comprises a number of data segments" as (page 551, lines 8-20).

As to claim 11, Weiss teaches the claimed limitations:

"a head representing a first pointer to a first leaf node; a tail representing a second pointer to a second leaf node" as a left binary tree in fig. 17.10, the rood node has two pointers, the first pointer points to left leaf node. The second pointer points to right leaf node. The first pointer of root node is presented as a head. The second pointer of root node is presented as a tail (page 516, lines 1-22). Weiss fails to teach the claimed limitation "a body, logically adjacent to the head and to the tail, having a set of pointers pointing to contiguous empty nodes". However, Weiss teaches as shown in fig. 17.10, a left binary tree has seven nodes, each node has two pointers to point to their leaf nodes. Several efficient implementations of priority queues use trees, BinaryNode (): left (NULL), righ (NULL). The above information shows that each leaf nodes can be empty node which stores no data (page 516, lines 1-22; page 517, lines 11-17).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Weiss's teaching of a left binary tree has seven nodes, each node has two pointers to point to their leaf nodes. Several efficient implementations of priority queues use trees, BinaryNode (): left (NULL), right (NULL) in order to store data in tree or sort data tree following user's desire.

As to claim 12, Weiss teaches the claimed limitation "wherein the redistribution process comprises the data structure traversing the tree in a first direction and a second direction" as (page 526, lines 1-10).

As to claim 13, Weiss teaches the claimed limitation "the first direction comprises a-logical, one, and the second direction comprises a logical zero" as (page 526, lines 1-9).

As to claim 14, Weiss teaches the claimed limitation "wherein the data structure traverses the tree by moving its head one leaf node towards its traveling direction" as (page 526, lines 1-9).

As to claim 15, Weiss teaches the claimed limitation "wherein the redistribution data structure traverses the tree structure in the first direction towards non-decreasing indices" as (page 526, lines 13-22; page 641, fig. 20.1).

As to claim 16, Weiss teaches the claimed limitation "wherein the redistribution data structure traverses the tree structure in the second direction towards non-increasing indices" as (page 526, lines 1-10; page 641, fig. 20.1).

As to claim 17, Weiss teaches the claimed limitation "wherein the redistribution data structure traverses the tree when a data segment is inserted and two conditions are met" as (page 648, lines 1-34).

As to claim 18, Weiss teaches the claimed limitation " wherein a first condition comprises a maximum threshold of filled spaces in the tree structure, and a second condition comprises a minimum threshold of filled spaces in the tree structure" as (page 644, lines 52-58; page 646, lines 5-8).

As to claim 19, Weiss teaches the claimed limitation "wherein the conditions are empirically determined" as (page 644, lines 52-58; page 646, lines 5-8).

As to claim 20, Weiss teaches the claimed limitation " wherein the redistribution data structure traverses the tree by moving one leaf node towards its traveling direction" as (page 550, lines 1-16).

As to claim 21, Weiss teaches the claimed limitation "an empty leaf node" as (page 550).

As to claim 22, Weiss teaches the claimed limitation "wherein certain conditions are met and the redistribution process continues" as (page 551, lines 1-21).

As to claims 23 and 43, Weiss teaches the claimed limitation "wherein the conditions are empirically calculated" as (page 551, lines 1-21).

As to claims 24 and 44, Weiss teaches the claimed limitation "wherein the redistribution process halts" (page 551, lines 1-21).

As to claims 25, 35 and 45, Weiss teaches the claimed limitation "wherein a data segment insertion restarts the redistribution process, and the traversal may continue where it was last halted" as (page 551, lines 1-21).

As to claim 26, Weiss teaches the claimed limitation "a non-empty leaf node" as if T=Null (page 551, line 10).

As to claim 27, Weiss teaches the claimed limitation "wherein the redistribution data structure copies the contents of its head into its tail" as(page 647, lines 1-11).

As to claims 28 and 36, Weiss teaches the claimed limitation "wherein the redistribution data structure travels towards non-decreasing indices" as (page 526, lines 1-10; page 641, fig. 20.1).

As to claims 29 and 39, Weiss teaches the claimed limitation "wherein the tree structure updates from leaf node level to root node level" as (fig. 20. 1, page 646).

As to claims 30 and 40, Weiss teaches the claimed limitation "wherein the contents of the head are cleared and the tail is moved a pre-calculated increment towards the traveling direction" as (page 650, lines 1-21).

As to claims 31 and 41, Weiss teaches the claimed limitation "wherein the increment is empirically determined" as (page 651, lines 1-22).

As to claims 32 and 42, Weiss teaches the claimed limitation "wherein certain conditions are met and the redistribution process continues" as (page 650, lines 1-21, page 651, lines 1-22).

As to claim 33, Weiss teaches the claimed limitation "wherein the conditions are empirically calculated" as (page 650, lines 1-21, page 651, lines 1-22).

As to claim 34, Weiss teaches the claimed limitation " wherein the redistribution process halts" as (page 651, lines 1-22).

As to claim 37, Weiss teaches the claimed limitation "wherein the tree structure updates between the tail and the nearest non-empty leaf node whose index is greater than the index of the tail" as fig. 20.14-20.15 and fig. 20.1( page 650; page 641).

As to claim 38, Weiss teaches the claimed limitation "changes made from leaf node level to root node level" as fig. 20.9 and fig. 20.10 (page 646).

As to claim 47, Weiss teaches the claimed limitation "wherein the process maintains the invariants of a sorted N-ary tree structure before and after the redistribution" as (pages 651, lines 5-22).

As to claim 48, Weiss teaches the claimed limitation "wherein the redistribution process maintains a consistent lookup operation on the sorted tree structure" as (page 648, lines 14-34).

4. Claim 46 is rejected under 35 U.S.C. 103(a) as being unpatentable over Mark ALLen Weiss, Algorithms, Data Structures, and Problem Solving With C++ 1996 in view of Leenstra, Sr. et al (USP 5303367).

As to claim 46, Weiss discloses the claimed limitation subject matter in claim 38, except the claimed limitation " wherein the tree structure may be reverse sorted". However, Weiss teaches that a tree is sorted in (fig. 20. 3, page 642). Also, Leenstra teaches that the linked Data Sets are maintained in inverted hierarchical sorted order relative to the Key Data Set at all times (fig. 10, col. 8, lines 35-42).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Leenstra's teaching of maintaining linked data sets in inverted hierarchical sorted order to Weiss's system in order to allow a user can sort a hierarchy data in any order.

***Conclusion***

5. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Troisi (USP 6385612).

***Contact Information***

*6.* Any inquiry concerning this communication or earlier communications from the examiner should be directed to Cam-Y Truong whose telephone number is (703-605-1169). The examiner can normally be reached on Mon-Fri from 8:00AM to 4:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kim Vu, can be reached on (703-305-4393). The fax phone numbers for the organization where this application or proceeding is assigned is (703)-746-7239 (formal communications intended for entry), or: (703)-746-7240 (informal communication labeled PROPOSED or DRAFT).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703-305-3900).

6/6/03

CY

SAlam
SHAHID AL ALAM
Primary PATENT EXAMINER